Application Note

Interfacing an HTML Form to the eZ80F91 MCU

AN020803-0708

Abstract

This application note demonstrates how to use Zilog's eZ80F91 microcontroller unit (MCU) as a web server to send electronic mails (emails) using an hypertext markup language (HTML) email submission form. This application note is an extension of another application note that describes sending emails using Zilog TCP/IP (ZTP) software suite. For more information, refer to *The eZ80F91 MCU as a Mail Server Application Note (AN0207)*.

Notes: 1. *The source code file associated with this application note,* `AN0208-SC01.zip`*, is available for download on* www.zilog.com.

2. *The source code files in this document are intended for use with the ZTP software suite v2.1.0 and ZDS II—eZ80Acclaim!® v4.11.0.*

Zilog® Product Overview

This section provides brief description of Zilog products used in this application note, which includes the award-winning Zilog's eZ80Acclaim*Plus!*™ MCUs and the full-featured Zilog TCP/IP (ZTP) software suite.

eZ80Acclaim*Plus!* MCU Family Overview

The eZ80Acclaim*Plus!* family of MCUs includes Flash and non-Flash products. The Flash-based eZ80Acclaim*Plus!* MCUs such as, eZ80F91, eZ80F92, and eZ80F93, are an exceptional value for designing high performance embedded applications. With speeds up to 50 MHz and an on-chip Ethernet MAC (eZ80F91 only), it has the performance necessary to execute complex applications supporting networking functions quickly and efficiently. Combining on-chip Flash and SRAM, eZ80Acclaim*Plus!* devices provide the memory required to implement communication protocol stacks and achieve flexibility when performing in-system updates of application firmware.

Zilog also offers two eZ80Acclaim*Plus!* devices without Flash memory—the eZ80L92 and eZ80190 microprocessors.

ZTP Overview

The ZTP integrates a rich-set of networking services with an efficient real-time operating system (RTOS). The operating system is a compact preemptive multitasking, multi threaded kernel with inter-process communications (IPC) support and soft real-time attributes. Table 1 lists the standard network protocols implemented as part of the embedded TCP/IP protocol stack in ZTP.

Table 1. Standard Network Protocols in ZTP

HTTP	TFTP	SMTP	Telnet	IP	PPP
DHCP	DNS	TIMEP	SNMP	TCP	UDP
ICMP	IGMP	ARP	RARP	FTP	PPPoE
SNTP	SSL*				

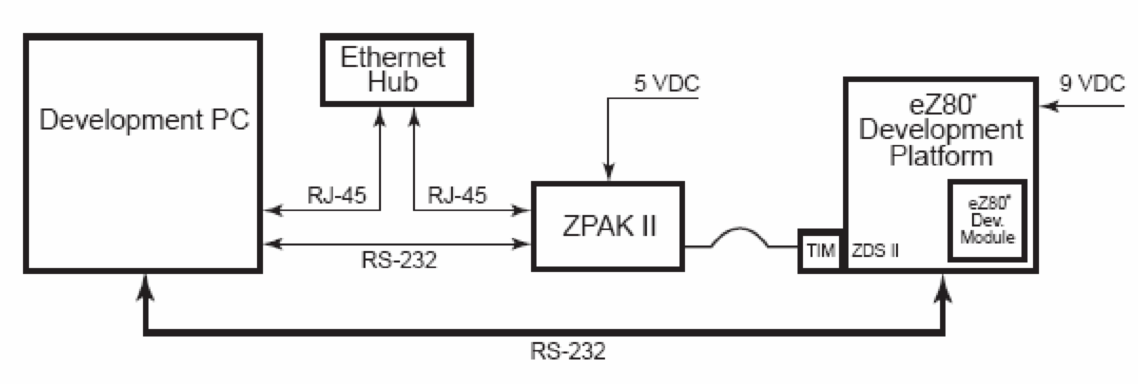
* SSL is available only with SSL package.

Many TCP/IP application protocols are designed using the client-server model. The final stack size is link-time configurable and determined by the protocols included in the build.

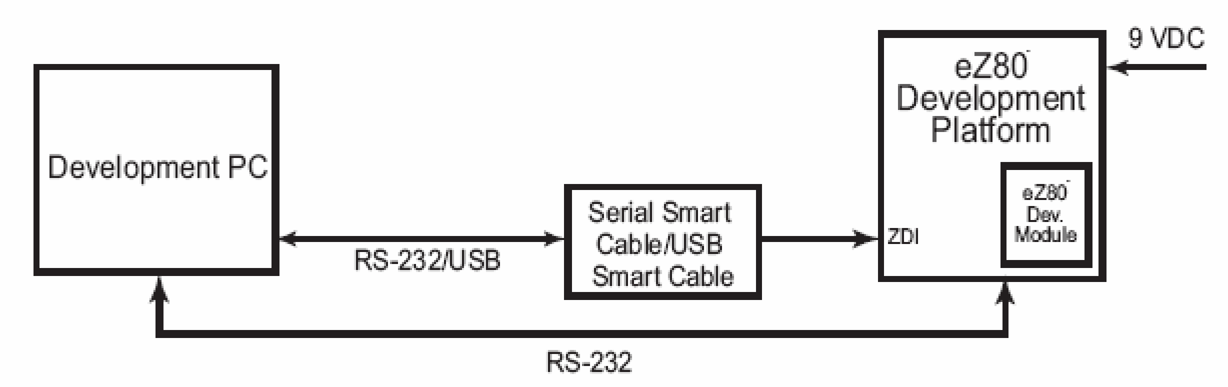
Discussion

Figure 1 displays the general data communication path for email using the eZ80F91 development kit, ZPAK II, and an Ethernet HUB. With this hardware, the eZ80F91 MCU can be used as an efficient web server when ZTP software is downloaded on to it. ZTP provides the software to drive the hardware used for TCP/IP connections. This hardware comprises of a SERIAL1 (UART1) device for PPP connections and the EMAC for Ethernet connections.

ZTP consists of a set of libraries that implement an embedded TCP/IP stack. ZTP functions only as a client, with a message body size of 32 K.



Hardware Setup for the ZTP Applications using ZPAK II



Hardware Setup for the ZTP Applications using Serial Smart Cable/USB Smart Cable

Figure 1. General Data Communication Path for Electronic Mail

ZTP API

The ZTP applications programming interface (API) allows to rapidly develop internet-ready applications with minimal effort by using any member of the eZ80® family of microprocessors (including the eZ80Acclaim*Plus!*™ product line). Because the API is common to all members of the eZ80 family, applications targeting one processor are easily ported to any other eZ80 devices.

For more details on the ZTP software suite, refer to *Zilog TCP/IP Software Suite Programmers Guide (RM0041)*. Downloading the ZTP software from www.zilog.com requires a registration key.

ZTP HTTP CGI Functions

Embedded systems do not contain a file system. Therefore, the embedded systems cannot save CGI scripts as separate programs.

Instead of saving CGI scripts as separate programs, ZTP uses C function calls, collectively called CGI functions. When a CGI function is called, it generates an HTML page that is sent to the browser. It is in these function calls that a code is written to read the information sent by a form via a web browser. This information is then processed as required by the application.

ZTP provides the following CGI functions.

- `int http_output_reply(http_request *request, int reply)`
- `char *http_find_argument (http_request *request, char *arg)`
- `int _http_write (http_request *request, char *buff, int count)`

In each ZTP CGI function, the pointer to the `request` structure keeps the requests from different clients separate.

The `http_output_reply()` function returns an acknowledgement to the browser that made the request.

The `http_find_argument()` function extracts parameters from the received data in the parsed browser request.

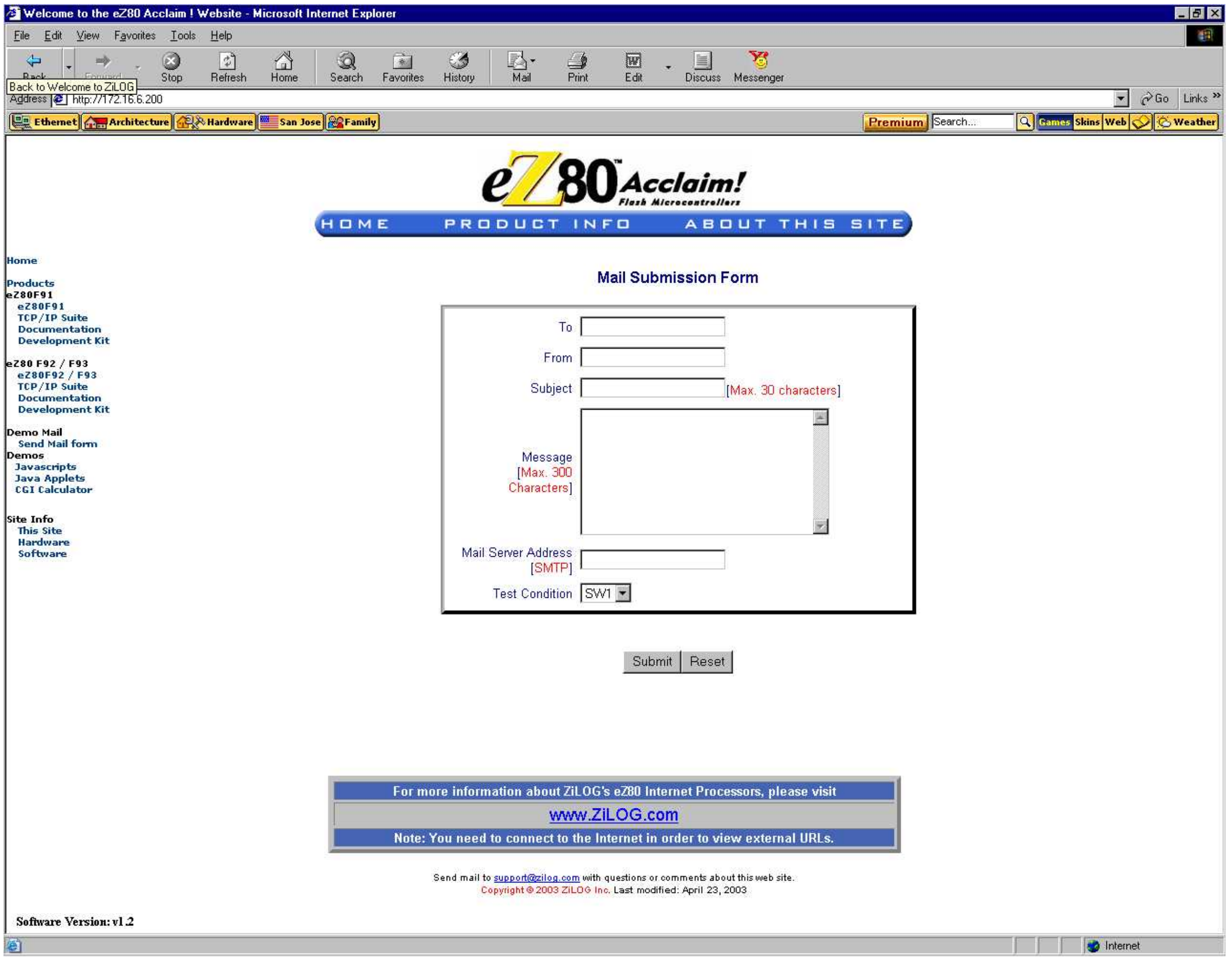
The `_http_write()` macro returns data to the browser that sent the request, that invoked the CGI function.

Developing the Form Interface Application

The mail server application described in *The eZ80F91 MCU as a Mail Server application note (AN0207)* requires the settings for the email to be entered directly into the `mail()` function. In the Form Interface application, this functionality is abstracted into an HTML form. The software implementation focuses on the functions developed for setting and storing the email content via an HTML submission form.

Reading Information from the Mail Submission Form

The email submission form displayed in Figure 2 on page 4 is an HTML form, generated from the `mailsubmission_form.htm` file. This file is located in the `AN0208-SC01.zip` file available for download on www.zilog.com.


Welcome to the eZ80 Acclaim ! Website - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss Messenger
Back to Welcome to ZiLOG
Address http://172.16.6.200
Go Links
Ethernet Architecture Hardware San Jose Family
Premium Search... Games Skins Web Weather
eZ80 Acclaim! Flash Microcontrollers
HOME PRODUCT INFO ABOUT THIS SITE
Home
Products
eZ80F91
eZ80F91
TCP/IP Suite
Documentation
Development Kit
eZ80 F92 / F93
eZ80F92 / F93
TCP/IP Suite
Documentation
Development Kit
Demo Mail
Send Mail form
Demos
Javascripts
Java Applets
CGI Calculator
Site Info
This Site
Hardware
Software
Submit
Reset
Internet

Mail Submission Form

To

From

Subject [Max. 30 characters]

Message [Max. 300 Characters]

Mail Server Address [SMTP]

Test Condition SW1

For more information about ZiLOG's eZ80 Internet Processors, please visit

www.ZiLOG.com

Note: You need to connect to the Internet in order to view external URLs.

Send mail to support@zilog.com with questions or comments about this web site.
Copyright © 2003 ZiLOG Inc. Last modified: April 23, 2003

Software Version: v1.2

Figure 2. Web Page with the Email Submission Form

The email submission form (see Figure 3 on page 5) consists of the following entry fields:

To—This field consists of the email ID destination in the *user@domainname.com* format.

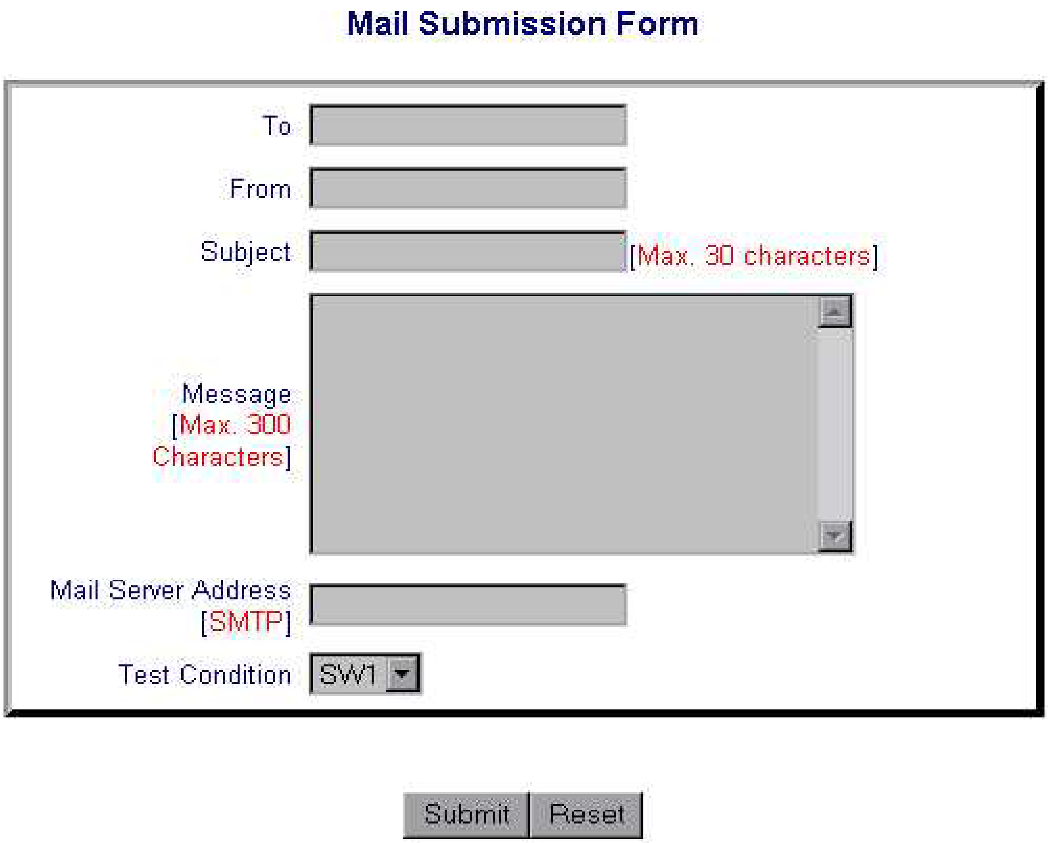
From—This field consists of the sender's email ID in the *eZ80webserver@zilog.com* format.

Subject—This field consists of the subject or topic of the email.

Message—This field consists of the text content of the message to be sent.

Mail Server Address—This field consists of the IP address of the SMTP mail server that distributes the email to the destination ID.

Test Condition—In this field, selecting a test condition from the drop down list links this email to a particular triggering event. Therefore, if the SW1 test condition is selected, this email is sent when the SW1 switch is pressed on the eZ80® Development Platform.



Mail Submission Form

To

From

Subject [Max. 30 characters]

Message [Max. 300 Characters]

Mail Server Address [SMTP]

Test Condition SW1

Figure 3. The Email Submission Form

The `mail_cgi()` function, located in the `mail_cgi.c` file, receive arguments from the mail submission HTML form and stores them in the `email_parm` structure, located in the `send_mail.h` file.

The Form Interface software is implemented for storing the contents of *three* emails in a single client database (formed by the `email_parm` structure). You must fill the email submission form and submit it. The submitted data (email content) corresponding to the selected switch is stored in the client database. When an event is triggered, (in this Demo, the triggering event is pressing the SW1/ SW2/SW3 switch on the eZ80® Development Platform) the corresponding email is fetched from the client database and sent to the specified mailbox.

The email submission form is also used for updating or modifying the client database.

Sending an Email

The `sendmail()` function (located in the `send_mail.c` file) is responsible for sending the email to the specified email ID. The standard ZTP `mail()` function is called within the `sendmail()` function.

For detailed descriptions on the `sendmail()` and `mail()` functions, refer to *The eZ80F91 MCU as a Mail Server application note (AN0207).*

Monitoring the Trigger Event to Send an Email

The SW1, SW2, and SW3 switches are used to generate the trigger events to the MCU to send the emails. The `init()` function (located in the `Test_cont.c` file) initializes the appropriate port pins as inputs to obtain the status of the switches.

Pressing the switch (SW1/SW2/SW3) on the eZ80® Development Platform causes the appropriate email to be sent to the required destination. The `Test_switch()` function (located in the `Test_cont.c` file) polls the SW1/SW2/SW3 switch continuously and reads the switch status in the main program. If any of the switches are pressed (logic 0), the `Test_switch()` function calls another function, `sendmail()` to send the mail to the specified address.

Note: *Any other trigger event may be used instead of switches.*

Adding and Integrating Form Interface Files to ZTP

The Form Interface application described in this application note requires the eZ80Acclaim*Plus!*™ Development Board, and ZTP stack. To execute this application, the files specific to the application must be added and integrated to the ZTP stack, before it is downloaded to the eZ80Acclaim*Plus!* Development Board. This section provides information on adding the Form Interface application files to the ZTP stack.

The Form Interface application files that must be added to the ZTP project files are located in the `AN0208-SC01.zip` file available for download on www.zilog.com.The files are of the following types:

- `C (*.c)` files
- `Header (*.h)` files
- `HTML (*.htm)` files

The ZTP stack is available on www.zilog.com and can be downloaded to a PC with a user registration key. ZTP can be installed in any specified location; the default location is:

```
C:\Program Files\Zilog
```

Note: *For information on ZDS II version and ZTP version used in this application, see* Hardware and Software Requirements *on page 8.*

Follow the steps below to add and integrate the application files to the ZTP stack:

1. Download ZTP (available for download on www.zilog.com). Browse to the location where ZTP is downloaded and open the `..\ZTP_2.1.0_Lib_ZDS\ZTP\SamplePrograms\ZTPDemo` folder.
2. Download the `AN0208-SC01.zip` file from www.zilog.com and extract the `.zip` file contents to a folder on your PC (this folder is referred to as `\AN0208-SC01` folder in this application note). The two extracted folders within the `\AN0208-SC01` folder are `\TD_Demo` and `\TD_Website.Acclaim`.
3. Copy the `mailsubmission_form.htm` file in the `\AN0208- SC01\TD_Website.Acclaim` folder to the `..\ZTP_2.1.0 Lib ZDS\ZTP \SamplePrograms\website.Acclaim` folder.
4. Copy all the `*.c` and `*.h` files located in the `AN0208-SC01\TD_Demo` folder to the `..\ZTP 2.1.0 Lib ZDS\ZTP\SamplePrograms\ZTPDemo` folder.
5. Launch ZDS II for eZ80Acclaim*Plus!* and open the `website.zdsproj project` located in the path `..\ZTP_2.1.0 Lib ZDS\ZTP\SamplePrograms\website.Acclaim.`
6. Click **Project → Add Files** to add the `mailsubmission_form.htm` file to the `website.zdsproj` project.

7. Open the `website.c` file from within ZDS II, and enter the following prototype declarations:

```
//prototype declarations
extern struct staticpage
mailsubmission_form_htm;
extern int mail_cgi(int);
```

8. The `website.c` file contains the array, `Webpage website[]`, which provides information on the HTML pages. Replace the last line of the array, `{0, NULL, NULL, NULL}` with the following lines of code:

```
{HTTP_PAGE_STATIC,
"/mailsubmission_form.htm","text/
html", &mailsubmission_form_htm},

{HTTP_PAGE_DYNAMIC, "/cgi-bin/mail",
"text/html", (struct
staticpage*)mail_cgi}, {0, NULL,
NULL, NULL}
```

9. From within ZDS II, open the `left.htm` file located in the `\Web Files` folder. Search for the `Demos <br>` statement and locate the `Demos <br>` code.

10. Add the following HTML lines of code above the `Demo <br>` statement in the `left.htm` file, to create a link from the eZ80Acclaim*Plus!*™ home page to the *Form Interface* web page, and the *Mail Submission* form.

```
Demo Mail<br>
&nbsp;&nbsp;&nbsp; <a
href="mailsubmission_form.htm"
target="main">Send Mail form</a><br>
```

11. Build the `website.zdsproj` project to obtain the new library file, `Acclaim_website.lib`.

12. Close the `website.zdsproj` project.

13. From within ZDS II, open the `ZTPDemo_F91.zdsproj` project available in the `..\ZTP_2.1.0 Lib ZDS\ZTP\SamplePrograms\ZTPDemo` filepath.

14. Click **Project → Add Files** to add all the `*.c` files located in the `..\AN0208-SC01\TD_Demo` folder to the `ZTPDemo_F91.zdsproj` project. The `*.c` files to be added are listed below:

- `mail_cgi.c`
- `send_mail.c`
- `Test_cont.c`

15. Open the `main.c` file of the `ZTPDemo_F91.zdsproj` project, and add the following include file:
`#include "send_mail.h"`

16. Check for the `commonServers` structure definition in the `ZTPConfig.c` file.

```
struct commonServers csTbl={
    "172.16.6.28",// Default Timer
                  //Server
    "",           // Default rfs
                  //server
    "", // Default File Server - N
     // not currently Used
    "172.16.6.194"// Default Name
                  //Server
};

struct If ifTbl[MAX_NO_IF]= {

// interface 0 -> Ethernet
//Configuration
{
    &usrDevBlk[0],// Control block
                  //for this device
    ETH,          // interface type
    ETH_MTU,      // MTU
    ETH_100,      // Speed can be
                  //ETH_10,AUTOSENCE
    "172.16.6.198",// Default IP
                  //address
    "172.16.6.1", // Default Gateway
    0xffff0000UL  // Default Subnet
                  //Mask

}
```

The `Bootrecord` variable contains network parameters and settings (in the four-octet dotted decimal format) specific to the local area network (LAN) at Zilog® as default. Modify the above structure definition with appropriate IP addresses within your local area network.

17. Add the following lines of code above the `return(OK)` statement at the end of the `INT16 ZTPAppEntry(void)` function:

```
Init();
while(1)
{
    Test_switch();
    RZKSuspendThread(RZKGetCurrentThr
    ead(),1);
}
```

18. Open the `eZ80HWConf.c` file and change the default MAC address (provided by ZTP) such that each eZ80® Development Platform on the LAN contains a unique MAC address. For example:

    ```
    const CHAR
    f91_mac_addr[ETHPKT_ALEN] =
    {0x00, 0x90, 0x23, 0x00, 0xDF,
    0x91};
    ```

    In the 6 byte MAC address shown above, the first 3 bytes must not be modified; the last 3 bytes can be used to assign a unique MAC address to the eZ80 Development Platform.

19. For this application, Dynamic Host Configuration Protocol (DHCP) is disabled; therefore, ensure that `byte b_use_dhcp = FALSE`.

20. Save the files and close the `ZTPDemo_F91` project.

Using the eZ80F91 Form Interface Application

This section provides the hardware and software requirements, the settings and the instructions to setup and execute the Form Interface application.

Hardware and Software Requirements

This section provides the hardware and software requirements for running the Form Interface application.

Hardware

The hardware required to execute the Form Interface application include:

- eZ80F91 Development Kit (eZ80F910200ZCO, eZ80F910300ZCO)
- PC with an Internet Browser and HyperTerminal

Software Requirements

The software required to execute the Form Interface application include:

- Zilog Developer Studio II—IDE for eZ80Acclaim!® v4.11.0
- Zilog TCP/IP Software Suite v2.1.0

Setup

The basic setup to assemble the Form Interface application Demo is displayed in Figure 1 on page 2. This setup displays the connections between the PC, LAN/WAN/Internet, and the eZ80F91 Development Kit.

Settings

The HyperTerminal and Jumper settings include:

HyperTerminal Settings

Set the HyperTerminal to 57.6 kbps Baud, 8-N-1, with no flow control

Jumper Settings

The Jumper settings for the eZ80 Development Platform include:

- J11, J7, and J2 are ON
- J3, J20, J21, and J22 are OFF
- For J14, connect 2 and 3
- For J19, MEM_CEN1 is ON, and CS_EX_IN, MEM_CEN2, and MEM_CEN3 are OFF

The Jumper settings for the eZ80F91 Module mounted on the eZ80 Development Platform include JP3 is ON

Procedure

The following steps describe the procedure to build and execute the Form Interface application.

1. Ensure that the required Form Interface application files are added and integrated to ZTP. For more information, see Adding and Integrating Form Interface Files to ZTP on page 6.
2. Connect the hardware as displayed in Figure 1 on page 2. Follow the jumper settings provided in Jumper Settings on page 8.
3. Connect the 5 V power supply to ZPAK II and the 7.5 V power supply to the Ethernet HUB.
4. Launch the HyperTerminal and follow the settings provided in the HyperTerminal Settings on page 8.
5. From within the HyperTerminal, press *z* repeatedly, and then press the **Reset** button on ZPAK II to view the menu to set the ZPAKII IP address.
6. Enter *H* to display help menu, and follow the menu instructions to obtain the IP address for ZPAK II in order to download the demo application file. This ZPAK II IP address must be entered in the ZDS II—IDE.
7. Launch ZDSII for eZ80Acclaim*Plus!*™ and open the Form Interface application project file (`ZTPDemo_F91.zdsproj`) located in the filepath `..\ZTP_2.1.0 Lib ZDS\ZTP\SamplePrograms\ZTPDemo`.
8. Open the `ZTPConfig.c` file. Ensure that the `BootInfo(struct commonServers, struct If ifTbl)` structure contains information that is relevant to your network configuration. Use the IP address in the structure to view the Form Interface web pages.
9. Build the project and download the resulting file to the eZ80F91 module mounted on the eZ80® Development Platform, using ZDS II. For details on the downloading procedure, refer to *Zilog TCP/IP Software Suite Programmers Guide (RM0041)*.
10. Execute the Form Interface application. For details on executing the Form Interface application, see Executing the Form Interface Application.

Executing the Form Interface Application

Follow the steps below to execute the Form Interface application:

1. Launch an Internet Browser and enter the server IP address in the URL field to open the eZ80Acclaim*Plus!*™ **Demo** home page.
2. Click the **Send Mail form** link in the left panel. The email submission form is displayed (see Figure 2 on page 4).
3. Enter the required information in all the fields. Select SW1 in the test condition field and click the **Submit** button to submit the form.
4. Press SW1 switch on the eZ80 Development Platform for one second.
5. Check the specified mailbox after a couple of minutes (depending on the network traffic) to see if an email with the predefined text is received.

The eZ80Acclaim*Plus!*—ZTP Mail Server successfully sends the email to the specified address.

Note: *Follow the same procedure to submit emails with test conditions SW2 and SW3. These emails are sent when the SW2 and SW3 switches are pressed.*

Summary

This application note highlights an HTML form interface to the eZ80F91 web server (eZ80F91 MCU with ZTP). An HTML form is a convenient and easy to use interface with the clearly defined fields to obtain specific input.

In this application note an HTML form is used to obtain and store email messages, that are sent only when specific events occur. Such a concept can be effectively used in a control system application, where concerned personnel receive timely alerts on pre-defined conditions.

References

The documents associated with eZ80® CPU, eZ80F91 MCU, ZPAK II, ZDS II for eZ80Acclaim!® MCUs, and ZTP available on www.zilog.com are provided below:

- eZ80® CPU User Manual (UM0077)
- eZ80F91 Development Kit User Manual (UM0142)
- eZ80F91 Product Specification (PS0192)
- eZ80F91 Module Product Specification (PS0193)
- The eZ80F91 MCU as a Mail Server Application Note (AN0207)
- ZPAK II Debug Interface Tool Product User Guide (PUG0015)
- Zilog Developer Studio II—eZ80Acclaim!® User Manual (UM0144)
- Zilog TCP/IP Stack API Reference Manual (RM0040)
- Zilog TCP/IP Software Suite Programmers Guide (RM0041)
- Zilog TCP/IP Software Suite Quick Start Guide (QS0049)

Warning: DO NOT USE IN LIFE SUPPORT

LIFE SUPPORT POLICY

ZILOG'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF ZILOG CORPORATION.

As used herein

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

Document Disclaimer

Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. ZILOG, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. ZILOG ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. The information contained within this document has been verified according to the general principles of electrical and mechanical engineering.

eZ80 and eZ80Acclaim*Plus!* are trademarks or registered trademarks of Zilog, Inc. All other product or service names are the property of their respective owners.